# Project documentation

## InsightStream - Navigate the News Landscape

### 1. Introduction

**Project Title:** InsightStream – Navigate the News Landscape
**Team ID:** NM2025TMID36502
**Team Leader:** NAVANEETHA Y [24csnavaneethay61@gmail.com]
**Team Members:**
– POOJA A[24cspooja067@gmail.com]
– ROHITHA P[24csrohita080@gmail.com]
– ROOBASRI M[24csroobasri081@gmail.com]

### 2. Project Overview

- **Purpose:**

    InsightStream empowers users to navigate the overwhelming news landscape by providing curated, bias-aware, and personalized news feeds. It helps readers discover diverse perspectives, track breaking news, and explore in-depth analyses in an intuitive interface.

- **Features:**

    – Personalized news recommendations

    – Bias and sentiment indicators for transparency

    – Topic-based news aggregation (Politics, Tech, Business, etc.)

    – Search and filter functionality (by region, category, or publisher)

    – Interactive dashboards for trend analysis

    – Secure user profiles with saved preferences

### 3. Architecture

- **Frontend**: React.js with Material UI & Tailwind for modern design

- **Backend**: Node.js + Express.js managing APIs and content aggregation

- **Database**: MongoDB storing user preferences, articles metadata, and analytics

- **Third-Party APIs**: News API, RSS feeds, sentiment analysis API

## 4. Setup Instructions

- **Prerequisites:**

  – Node.js

  – MongoDB

  – Git

  – React.js

  – Express.js

- **Installation Steps:**

  # Clone the repository

  git clone <repo-url> insightstream

  # Install client dependencies

  cd client

  npm install

  # Install server dependencies

  cd ../server

  npm install

## 5. Folder Structure

  InsightStream/

  |-- client/   # React frontend

```
|   |__ components/
|   |__ pages/
|-- server/   # Node.js backend
|   |__ routes/
|   |__ models/
|   |__ controllers/
```

## 6. Running the Application

- **Frontend:**

  cd client

  npm start

- **Backend:**

  cd server

  npm start

  Access: Visit http://localhost:3000

## 7. API Documentation

- **User:**

  POST /api/user/register

  POST /api/user/login

- **News:**

  GET /api/news/trending

  GET /api/news/category/:category

  GET /api/news/search?q=keyword

- **Preferences:**

POST /api/preferences/save

GET /api/preferences/:userId

## 8. Authentication

- WT-based authentication for secure user sessions

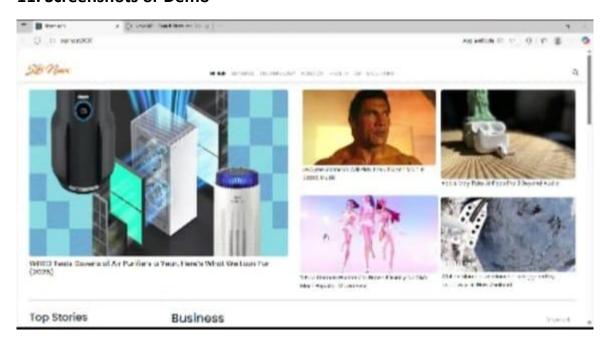- Role-based access control for admins

## 9. User Interface

- Landing Page (latest headlines, categories)

- Personalized Dashboard (recommended articles, bias insights)

- News Explorer (search, filter, analyze trends)

- Admin Panel (manage content & users)

## 10. Testing

- Manual Testing during sprints

- Tools: Postman (API), Chrome DevTools (UI), Jest (unit tests)

## 11. Screenshots or Demo

## 12. Known Issues

- Some RSS feeds may provide duplicate stories

- Sentiment analysis may occasionally misclassify nuanced text

## 13. Future Enhancements

- AI-powered summarization of long articles

- Cross-platform mobile app (React Native)

- Multilingual news aggregation

- Integration with social platforms for sharing